
TRS - 80 USERS GROUP NEWSLETTER

OCTOBER 1978 FAYETTEVILLE, NC VOLUME 1, NUMBER 9

```

10 CLS
15 PRINTTAB(10);">>> 23 MATCHES <<<"
20 PRINTTAB(5);"BY JOHN MARLER: SAN JOSE, CA"
30 FORI=1TO1000:NEXTI:CLS
1000 PRINT"LETS PLAY 23 MATCHES"
1010 PRINT"YOU MOVE FIRST-TAKE 1,2,OR 3 MATCHES"
1020 PRINT"THE TRS-80 WILL TAKE 1,2,OR 3 MATCHES"
1030 PRINT"TAKE TURNS MOVING-THE ONE WHO MUST TAKE"
1040 PRINT"THE LAST MATCH LOSES":INPUT"PUSH ENTER WHEN READY";A$
1050 CLS:M=23:FORD=3TO113STEP5:FORE=2TO14:SET(D,E):SET(D+1,E):NEXTE:NEXTD
1051 B=1
1060 FORE=2TO3:FORD=2TO112STEP5:SET(D,E):SET(D+3,E):NEXTD:NEXTE:D=3
1065 PRINTAT320,"MOVE", "MATCHES LEFT", "YOUR MOVE", "TRS-80 MOVE"
1070 PRINTAT(0*64+320),0,M,:INPUTT
1075 IFT>32767GOTO1900
1090 IF(T*M)+(T<)INT(T)+(T<1)+(T>3)GOTO1900
1100 M=M-T:G=T-1:GOSUB1950:IFM=0GOTO1300
1110 IFM=1GOTO1400
1120 R=M-4*INT(M/4):IFR=1C=RND(3):GOTO1135
1130 C=(R+3)-4*INT((R+3)/4)
1135 PRINT0(0*64+320),0,M+T,T,C:0=B+1
1140 M=M-C:G=C-1:GOSUB1950:GOTO1070
1300 PRINT"YOU LOSE!!! DON'T FEEL TOO BAD - MOST DO LOSE":GOTO2000
1400 PRINT"YOU WON !!! - YOU'RE BETTER THAN THE AVERAGE PLAYER":GOTO2000
1900 PRINTAT(0*64+340),"INVALID MOVE":FORI=1TO1000:NEXTI:GOTO1070
1950 FORD=0TOG*5+DSTEP5:FORE=2TO14:FORZ=-1TO3:RESET(D+Z,E):NEXTZ
1960 NEXTE:NEXTD:RETURN
2000 INPUT"WINAT ANOTHER TRY (1=YES,2=NO)";X
2001 IFX=1THEN1050
2002 CLS:PRINT"THANKS - SEE 'YA AGAIN, SPORT!"
2003 FORI=1TO750:NEXTI:CLS:PRINT"GOODBYE"
2004 END
2110 FORX=3TO13STEP2:RESET(X,41):NEXTX:RESET(7,43):RESET(8,43)

```

GORDON,

HERE IS A SHORT PROGRAM THAT WILL IDENTIFY THE ASCII
CHARACTER CODE # AND PRINT A LINE OF THE CORRESPONDING
CHARACTERS. (FOR LEVEL II)

CODES FROM 1 TO 31 ARE CONTROL CODES. 32 TO 128 ARE
CHARACTER CODES. 129 TO 191 ARE GRAPHIC CODES. 192 TO 255
ARE SPACE COMPRESSION CODES.

THE GRAPHIC CODES (129 TO 191) ARE VERY INTERESTING.
USING THESE CODES SOME FANTASTIC GRAPHICS CAN BE MADE.

KEEP UP THE GOOD WORK WITH THE NEWSLETTER.

TERRY HAZELETT, 2107 CAPITOL DR., PARKERSBURG, WV 26101

```

210 CLS
220 FOR I=33 TO 191
230 FOR B=1 TO 500:NEXT B
240 PRINT "ASCII CODE #";I
250 FOR A=1 TO 64
260 PRINT CHR$(I);
270 NEXT A
280 NEXT I
290 END

```

ASCII CODE # 42

ASCII CODE # 43

3 Grants Lane
Ossining, N.Y. 10562
August 6, 1978

Dear Bob,

Enclosed is a tape which contains two programs which some of your readers may find useful. They are for converting LEVEL-1 data files to LEVEL-2. The program supplied with LEVEL-2 requires inter-changing tapes for each record. The enclosed programs will reduce the labor involved in converting large data files.

Program L1L2A is designed for single cassette systems. It will read a selected number of (up to 99) of LEVEL-1 records and store them in a buffer. The tape is then replaced and the buffer is dumped in LEVEL-2 format.

Program L1L2B is designed for use with the expansion unit and two tape recorders. It will repetitively read a level one record from one cassette and write to the other. The record count is displayed on the screen. If the checksum does not compare, an error message will follow the record number. Control returns to the beginning of the program after 99 records; at which point one can continue for another 99 records or return to LEVEL-2 BASIC.

The source and object files are both included and were generated with the R-S Editor-Assembler. The entry point for the assembled version is 5000 (20480 decimal) and the buffer starts at 5302. The programs can be relocated by changing the origin and stack locations in the first two source lines and then reassembling. The source statements are commented so that they may be modified for individual preference.

Sincerely,


Tom Michel

5000	00100	ORG	5000H	
5300	00110	STACK	EQU	5300H
002B	00120	KB	EQU	002BH ; KEYBOARD SCAN
0033	00130	VD	EQU	0033H ; DISPLAY (A)
01F0	00140	STOP	EQU	01F0H ; STOP CASSETTE DRIVE
0212	00150	DEFC	EQU	0212H ; DEFINE CASSETTE DRIVE
021E	00160	OUT	EQU	021EH ; OUTPUT (A) TO CAS. POK
0264	00170	WRT	EQU	0264H ; WRITE (A) ON TAPE
0287	00180	SYNC	EQU	0287H ; WRITE SYNC BYTE AS
1A19	00190	RETB	EQU	1A19H ; RETURN TO BASIC
5000	310053		LD	SP, STACK
5003	3E0E		LD	A, 0EH ; TURN ON CURSOR
5005	CD0251		CALL	VIDEO
5008	CDEC50		CALL	CLEAR
500B	217F51	BEG	LD	HL, M1
500E	0670		LD	B, 70H
5010	CDE450		CALL	PRINT ; PRINT MESSAGE 1
5013	CD5251		CALL	COUNT ; SET RECORD COUNTER
5016	CD3451		CALL	PNUM ; PRINT VALUE OF COUNTER
5019	21EF51		LD	HL, M2
501C	062A		LD	B, 2AH
501E	CDE450		CALL	PRINT ; PRINT MESSAGE 2

5021	211952	00320	LD	HL, M3	
5024	0649	00330	LD	0, 49H	
5026	CDE450	00340	CALL	PRINT	; PRINT MESSAGE 3
5029	0DF750	00350 J1	CALL	KEY	
502C	28F0	00360	JR	Z, J1	
502E	FE52	00370	CP	52H	
5030	28D6	00380	JR	Z, BEG	
5032	FE42	00390	CP	42H	
5034	CA191A	00400	JP	Z, RET0	
5037	210253	00410	LD	HL, STACK+2	; BUFFER ORIGIN=STACK+2
503A	1601	00420	LD	D, 01H	; INITIALIZE COUNTER
503C	1001	00422	JR	LOOP+1	
503E	14	00424 LOOP	INC	D	
503F	AF	00430	XOR	A	; READ LOOP
5040	7A	00440	LD	A, D	
5041	27	00450	DAA		
5042	57	00460	LD	D, A	
5043	CD3451	00470	CALL	PNUM	; DISPLAY RECORD COUNT
5046	AF	00480	XOR	A	
5047	CD1202	00490	CALL	DEFC	; SET CAS LATCH FOR #1
504A	AF	00500	XOR	A	
5040	CD0C51	00510 J2	CALL	CAS	; START CAS AND READ IN
504E	FEA5	00520	CP	0A5H	; BITS UNTIL (A)=A5 WHICH
5050	20F9	00530	JR	NZ, J2	; IS THE SYNC BYTE
5052	CD2C51	00540	CALL	CHR	; READ THE 1ST 4 BYTES
5055	CD2C51	00550	CALL	CHR	; AND DISREGARD
5058	CD2C51	00560	CALL	CHR	
5058	CD2C51	00570	CALL	CHR	
505E	0E00	00580	LD	C, 0	; INITIALIZE CHECKSUM
5060	CD2C51	00590 J3	CALL	CHR	; GET BYTE
5063	77	00600	LD	(HL), A	; STORE IN BUFFER
5064	23	00610	INC	HL	; INCREMENT POINTER
5065	F5	00620	PUSH	AF	
5066	01	00630	ADD	A, C	; COMPUTE CHECKSUM
5067	4F	00640	LD	C, A	
5060	F1	00650	POP	AF	
5069	FE0D	00660	CP	00H	; LOOK FOR END OF
5068	20F3	00670	JR	NZ, J3	; RECORD = 00
506D	CD2C51	00680	CALL	CHR	; LAST BYTE= NEG CHECKSUM
5070	01	00690	ADD	A, C	
5071	C2D950	00700	JP	NZ, ERR	
5074	3A0153	00720 J4	LD	A, (STACK+1)	
5077	BA	00730	CP	D	
5070	20C4	00740	JR	NZ, LOOP	
507A	0DF801	00760	CALL	STOP	
507D	216252	00770	LD	HL, M4	
5080	0627	00780	LD	0, 27H	
5082	CDE450	00790	CALL	PRINT	; PRINT MESSAGE 4
5085	211952	00800	LD	HL, M3	
5088	0649	00810	LD	0, 49H	
508A	CDE450	00820	CALL	PRINT	; PRINT MESSAGE 3
508D	0DF750	00830 J6	CALL	KEY	
5090	28F0	00840	JR	Z, J6	
5092	FE52	00850	CP	52H	
5094	CA0850	00860	JP	Z, BEG	
5097	FE42	00870	CP	42H	
5099	CA191A	00880	JP	Z, RET0	
509C	210253	00890	LD	HL, STACK+2	; BUFFER ORIGIN=STACK+2
509F	1601	00900	LD	D, 01H	; INITIALIZE COUNTER
50A1	1001	00905	JR	LOOP2+1	
50A3	14	00907 LOOP2	INC	D	
50A4	AF	00910	XOR	A	; WRITE LOOP
50A5	7A	00920	LD	A, D	
50A6	27	00930	DAA		

50A7	57	00940	LD	D,A	
50AB	CD3451	00950	CALL	PNUM	; PRINT COUNT
50AC	AF	00960	XOR	A	
50AC	CD1202	00970	CALL	DEFC	; SET CAS LATCH FOR #1
50AF	CD9702	00980	CALL	SYNC	; WRITE SYNC BYTE
50B2	7E	00990 J5	LD	A, (HL)	; WRITE THE CONTENTS OF
50B3	23	01000	INC	HL	; THE BUFFER. NO
50B4	CD6402	01010	CALL	WRT	; CHECKSUM IS USED IN
50B7	FE00	01020	CP	0DH	; LEVEL-II DATA
50B9	20F7	01030	JR	NZ, J5	
50BB	CDF801	01040	CALL	STOP	; STOP CASSETTE
50BE	3A0153	01060	LD	A, (STACK+1)	
50C1	BA	01070	CP	D	
50C2	20DF	01080	JR	NZ, LOOP2	
50C4	218952	01100	LD	HL, M5	
50C7	0650	01110	LD	B, 5BH	
50C9	CDE450	01120	CALL	PRINT	; PRINT MESSAGE 5
50CC	CD750	01130 J7	CALL	KEY	
50CF	20FB	01140	JR	Z, J7	
50D1	FE42	01150	CP	42H	
50D3	CA191A	01160	JP	Z, RETB	
50D6	C30850	01170	JP	BEG	
50D9	21E452	01180 ERR	LD	HL, M6	; ERR - DISPLAYS ON THE
50DC	0600	01190	LD	B, 0DH	; MONITOR THE MESSAGE
50DE	CDE450	01200	CALL	PRINT	; "INPUT ERROR"
50E1	C37450	01210	JP	J4	
50E4	7E	01220 PRINT	LD	A, (HL)	; PRINT - DISPLAYS ON THE
50E5	CD0251	01230	CALL	VIDEO	; THE MESSAGE POINTED TO
50E8	23	01240	INC	HL	; BY (HL). THE MESSAGE
50E9	10F9	01250	DJNZ	PRINT	; LENGTH MUST FIRST BE
50EB	C9	01260	RET		; LOADED INTO THE B REG
50EC	3E1C	01270 CLEAR	LD	A, 1CH	; POSITION CURSOR TO
50EE	CD0251	01280	CALL	VIDEO	; HOME(00)
50F1	3E1F	01290	LD	A, 1FH	; CLEAR TO END OF FRAME
50F3	CD0251	01300	CALL	VIDEO	
50F6	C9	01310	RET		
50F7	D5	01320 KEY	PUSH	DE	; KEY - SCANS KEYBOARD &
50F8	FDE5	01330	PUSH	IY	; RETURNS ASCII REPRESENTATION OF KEY RETURNED
50FA	CD2B00	01340	CALL	KB	
50FD	B7	01350	OR	A	; TO A. IF KEYBOARD IS
50FE	FDE1	01360	POP	IY	; CLEAR A=0. OR A IS USED
5100	D1	01370	POP	DE	; TO SET FLAGS
5101	C9	01380	RET		
5102	D5	01390 VIDEO	PUSH	DE	; VIDEO - DISPLAYS (A)
5103	FDE5	01400	PUSH	IY	; ON MONITOR
5105	CD3300	01410	CALL	VD	
5108	FDE1	01420	POP	IY	
510A	D1	01430	POP	DE	
510B	C9	01440	RET		
510C	D9	01450 CAS	EXX		; CAS - RETRIEVES ONE
510D	00	01460	EX	AF, AF'	; BIT FROM THE CASSETTE
510E	DBFF	01470 J0	IN	A, (0FFH)	; PORT - FF
5110	17	01480	RLA		
5111	30FB	01490	JR	NC, J0	
5113	067C	01500	LD	0, 7CH	
5115	10FE	01510 J9	DJNZ	J9	
5117	CD1E02	01520	CALL	OUT	
511A	06F0	01530	LD	0, 0F8H	
511C	10FE	01540 J10	DJNZ	J10	
511E	DBFF	01550	IN	A, (0FFH)	
5120	47	01560	LD	0, A	
5121	08	01570	EX	AF, AF'	
5122	C010	01580	RL	B	
5124	17	01590	PLA		

5125 F5	01600	PUSH	AF
5126 CD1E02	01610	CALL	OUT
5129 F1	01620	POP	AF
512A 09	01630	EXX	
512B 09	01640	RET	
512C 0608	01650	LD	B, 8H
512E CD0C51	01660	J11	CAS
5131 10F8	01670	DJNZ	J11
5133 09	01680	RET	
5134 7A	01690	PNUM	LD A, D
5135 0604	01700	LD	B, 04H
5137 083F	01710	J13	SRL A
5139 10FC	01720	DJNZ	J13
513B 4F	01730	LD	C, A
513C 3E30	01740	LD	A, 30H
513E 81	01750	ADD	A, C
513F CD0251	01760	CALL	VIDEO
5142 7A	01770	LD	A, D
5143 E60F	01780	AND	0FH
5145 4F	01790	LD	C, A
5146 3E30	01800	LD	A, 30H
5148 81	01810	ADD	A, C
5149 CD0251	01820	CALL	VIDEO
514C 3E20	01830	LD	A, 20H
514E CD0251	01840	CALL	VIDEO
5151 09	01850	RET	
5152 0602	01860	COUNT	LD B, 02H
5154 0E00	01870	LD	C, 0H
5156 1600	01880	LD	D, 0H
5158 CDF750	01890	J12	CALL KEY
515B 20F8	01900	JR	Z, J12
515D 4F	01910	LD	C, A
515E C606	01920	ADD	A, 0C6H
5160 38F6	01930	JR	C, J12
5162 79	01940	LD	A, C
5163 D630	01950	SUB	30H
5165 FA5851	01960	JP	M, J12
5168 4F	01970	LD	C, A
5169 78	01980	LD	A, 8
516A FE01	01990	CP	01H
516C 2804	02000	JR	Z, J15
516E 79	02010	LD	A, C
516F 57	02020	LD	D, A
5170 10E6	02030	DJNZ	J12
5172 7A	02040	J15	LD A, D
5173 0604	02050	LD	B, 04H
5175 C827	02060	SLA	A
5177 10FC	02070	DJNZ	J15+3
5179 81	02080	ADD	A, C
517A 57	02110	LD	D, A
517B 320153	02120	LD	(STACK+1), A
517E 09	02130	RET	

; PNUM - PRINTS VALUE OF
 ; COUNTER (D REG) THE
 ; UPPER & LOWER NIBBLE
 ; REPRESENT A TWO DIGIT
 ; DECIMAL #

J15	5172
J12	5158
J13	5137
J11	512E
J10	511C
J9	5115
J8	510E
M6	52E4
J7	50CC
M5	5209
J5	5082

; COUNT - GETS TWO DIGITS
 ; EACH 0-9 FIRST DIGIT
 ; IS SHIFTED INTO THE
 ; UPPER NIBBLE & RESULT
 ; IS STORED IN THE D REG
 ; & IN STACK+1

LOOP2	50A3	COUNT	5152
J6	508D	PRINT	50E4
M4	5262	M1	517F
J4	5074	CLEAR	50EC
ERR	50D9	BEG	5088
J3	5060	VIDEO	5102
CHR	512C	RET8	1A19
CAS	510C	SYNC	0287
J2	5048	WRT	0264
LOOP	503E	OUT	021E
KEY	50F7	DEFC	0212
J1	5029	STOP	01F8
M3	5219	VD	0033
M2	51EF	KB	0020
PNUM	5134	STACK	5300

02140 M1	DEFM	' LEVEL-1 TO LEVEL-2 DATA CONVERSION'
02150	DEFW	0000H
02160	DEFM	'TYPE THE # OF RECORDS, 00-99, TO 8E'
02170	DEFB	0DH
02180	DEFM	'CONVERTED YOU MUST TYPE TWO DIGITS'
02190	DEFW	3F00H
02200 M2	DEFW	0000H
02210	DEFM	'LOAD THE LEVEL-1 TAPE, PUSH PLAY, & HIT'
02220	DEFB	0DH
02230 M3	DEFM	'ENTER - TO CONTINUE'
02240	DEFB	0DH
02250	DEFM	'R - TO RESTART

02260	DEFB	0DH	
02270	DEFM	'B	- TO RETURN TO BASIC'
02280	DEFB	0D0DH	
02290 M4	DEFB	0DH	
02300	DEFM	'LOAD LEVEL-2 TAPE, PUSH RECORD, & HIT'	
02310	DEFB	0DH	
02320 M5	DEFB	0DH	
02330	DEFM	'	CONVERSION COMPLETE'
02340	DEFB	0D0DH	
02350	DEFM	'TYPE R TO CONVERT MORE RECORDS'	
02360	DEFB	0DH	
02370	DEFM	'OR 0 TO RETURN TO BASIC'	
02380	DEFB	3F0DH	
02390 M6	DEFM	' INPUT ERROR '	

5000	00100	ORG	5000H	
5300	00110 STACK	EQU	5300H	
0020	00120 K0	EQU	002BH	; KEYBOARD SCAN
0033	00130 VD	EQU	0033H	; DISPLAY (A)
01F0	00140 STOP	EQU	01F0H	; STOP CASSETTE DRIVE
0212	00150 DEFC	EQU	0212H	; DEFINE CASSETTE DRIVE
021E	00160 OUT	EQU	021EH	; OUTPUT (A) TO CAS. PORT
0264	00170 WRT	EQU	0264H	; WRITE (A) ON TAPE
0287	00180 SYNC	EQU	0287H	; WRITE SYNC BYTE A5
1A19	00190 RETB	EQU	1A19H	; RETURN TO BASIC
5000 310053	00200	LD	SP, STACK	
5003 3E0E	00210	LD	A, 0EH	; TURN ON CURSOR
5005 CD8050	00220	CALL	VIDEO	
5000 CDA750	00230 BEG	CALL	CLEAR	
5000 211051	00240	LD	HL, M1	
500E 069D	00250	LD	0, 9DH	
5010 CD9F50	00260	CALL	PRINT	; PRINT MESSAGE 1
5013 CDB250	00270 J1	CALL	KEY	
5016 20F8	00280	JR	Z, J1	
5010 FE42	00290	CP	42H	; IF KEYED INPUT IS 0
501A CA191A	00300	JP	Z, RETB	; RETURN TO BASIC
501D 3E0D	00310	LD	A, 0DH	
501F CD8050	00320	CALL	VIDEO	
5022 3E01	00330	LD	A, 01	
5024 320153	00340	LD	(STACK+1), A	; INITIALIZE RECORD COUNT
5027 210253	00350 LOOP	LD	HL, STACK+2	; BUFFER ORIGIN=STACK+2
502A CDEF50	00360	CALL	PNUM	; DISPLAY RECORD COUNT
502D 3EFF	00370	LD	A, 0FFH	
502F CD1202	00380	CALL	DEFC	; SET CAS LATCH FOR #2
5032 AF	00390	XOR	A	
5033 CDC750	00400 J2	CALL	CAS	; START CAS AND READ IN
5036 FEAS	00410	CP	0A5H	; 01TS UNTIL (A)=THE SYNC
5030 20F9	00420	JR	NZ, J2	; BYTE - A5.
503A 3E2A	00430	LD	A, 2AH	; 2A = ASCII *
503C 323E3C	00440	LD	(3C3EH), A	; DISPLAY *
503F 323F3C	00450	LD	(3C3FH), A	
5042 CDE750	00460	CALL	CHR	; READ THE 1ST 4 BYTES
5045 CDE750	00470	CALL	CHR	; AND DISREGARD
5040 CDE750	00480	CALL	CHR	
5048 CDE750	00490	CALL	CHR	
504E 0E00	00500	LD	C, 0	; INITIALIZE CHECKSUM
5050 CDE750	00510 J3	CALL	CHR	; GET BYTE
5053 77	00520	LD	(HL), A	; STORE IN BUFFER
5054 23	00530	INC	HL	; INCREMENT POINTER
5055 F5	00540	PUSH	AF	
5056 B1	00550	ADD	A, C	; COMPUTE CHECKSUM
5057 4F	00560	LD	C, A	
5050 F1	00570	POP	AF	
5059 FE0D	00580	CP	0DH	; LOOK FOR END OF

5050 20F3	00530	JR	NZ, J3	RECORD = 00
5050 C0E750	00600	CALL	CHK	LAST BYTE= NEG CHECKSUM
5060 81	00610	ADD	A, C	
5061 C29450	00620	JP	NZ, ERF	
5064 C0F801	00630 J14	CALL	STOP	STOP CAS #2
5067 3E20	00640	LD	A, 20H	
5069 323F30	00650	LD	(303FH), A	TURN OFF +
506C 210253	00660	LD	HL, STACK+2	BUFFER ORIGIN=STACK+2
506F AF	00670	XOR	A	
5070 C01202	00680	CALL	DEFC	SET CAS LATCH FOR #1
5073 C08702	00690	CALL	SYNC	WRITE SYNC BYTE
5076 7E	00700 J5	LD	A, (HL)	WRITE THE CONTENTS
5077 23	00710	INC	HL	OF THE BUFFER. NO
5078 C06402	00720	CALL	WRT	CHECKSUM IS USED IN
507B FE0D	00730	CP	0DH	LEVEL-11 DATA
507D 20F7	00740	JR	NZ, J5	
507F C0F801	00750	CALL	STOP	STOP CASSETTE #1
5082 3A0153	00760	LD	A, (STACK+1)	
5085 3C	00770	INC	A	INCREMENT & DECIMAL
5086 27	00780	DAA		ADJUST & STORE THE
5087 320153	00790	LD	(STACK+1), A	THE RECORD COUNTER
508A FE99	00800	CP	99H	
508C CA0850	00810	JP	Z, BEG	
508F 1896	00820	JR	LOOP	
5091 C3191A	00830	JP	RET8	
5094 21AD51	00840 ERR	LD	HL, M2	ERR - DISPLAYS ON THE
5097 060D	00850	LD	8, 0DH	MONITOR THE MESSAGE
5099 C09F50	00860	CALL	PRINT	"INPUT ERROR"
509C C36450	00870	JP	J14	
509F 7E	00880 PRINT	LD	A, (HL)	PRINT - DISPLAYS
50A0 C0BD50	00890	CALL	VIDEO	THE MESSAGE POINTED TO
50A3 23	00900	INC	HL	BY (HL). THE MESSAGE
50A4 10F9	00910	DJNZ	PRINT	LENGTH MUST FIRST BE
50A6 C9	00920	RET		LOADED INTO THE 8 REG.
50A7 3E1C	00930 CLEAR	LD	A, 1CH	POSITION CURSOR TO
50A9 C0BD50	00940	CALL	VIDEO	HOME(00)
50AC 3E1F	00950	LD	A, 1FH	CLEAR TO END OF
50AE C0BD50	00960	CALL	VIDEO	FRAME
50B1 C9	00970	RET		
50B2 D5	00980 KEY	PUSH	DE	KEY - SCANS KEYBOARD &
50B3 FDE5	00990	PUSH	IY	RETURNS ASCII REPRESENTATION OF KEY RETURNED
50B5 C02B00	01000	CALL	KB	TO A. IF KEYBOARD IS
50B8 B7	01010	OR	A	CLEAR A=0 OR A IS USED
50B9 FDE1	01020	POP	IY	TO SET FLAGS
50BB D1	01030	POP	DE	
50BC C9	01040	RET		
50BD D5	01050 VIDEO	PUSH	DE	VIDEO - DISPLAYS (A)
50BE FDE5	01060	PUSH	IY	PRESERVES REGISTER
50C0 C03300	01070	CALL	VD	CONTENTS
50C3 FDE1	01080	POP	IY	
50C5 D1	01090	POP	DE	
50C6 C9	01100	RET		
50C7 D9	01110 CAS	EXX		CAS - RETRIEVES ONE
50C8 08	01120	EX	AF, AF'	BIT FROM THE CASSETTE
50C9 D0FF	01130 J8	IN	A, (0FFH)	PORT - FF
50CB 17	01140	RLA		
50CC 30FB	01150	JR	NC, J8	
50CE 067C	01160	LD	B, 7CH	
50D0 10FE	01170 J9	DJNZ	J9	
50D2 C01E02	01180	CALL	OUT	
50D5 06F8	01190	LD	B, 0FEH	
50D7 10FE	01200 J10	DJNZ	J10	
50D9 D0FF	01210	IN	A, (0FFH)	
50DB 47	01220	LD	B, A	
50DC 08	01230	EX	AF, AF	

50D0	0B10	01240	RL	B			
50D1	17	01250	RLA				
50E0	F5	01260	PUSH	AF			
50E1	CD1E02	01270	CALL	OUT			
50E4	F1	01280	POP	AF			
50E5	09	01290	EXX				
50E6	C9	01300	RET				
50E7	0608	01310	CHR	LD	B, 8H		
50E9	CD0750	01320	J11	CALL	CAS		
50EC	10F0	01330	DJNZ	J11			
50EE	C9	01340	RET				
50EF	3A0153	01350	PNUM	LD	A, (STACK+1)		; PNUM - DISPLAY THE
50F2	57	01360	LD	D, A			; VALUE OF THE RECORD
50F3	0604	01370	LD	8, 04H			; COUNTER THAT IS
50F5	C03F	01380	J13	SRL	A		; STORED IN STACK+1
50F7	10FC	01390	DJNZ	J13			
50F9	4F	01400	LD	C, A			
50FA	3E30	01410	LD	A, 30H			
50FC	01	01420	ADD	A, C			
50FD	CD0D50	01430	CALL	VIDEO			
5100	7A	01440	LD	A, D			
5101	E60F	01450	AND	0FH			
5103	4F	01460	LD	C, A			
5104	3E30	01470	LD	A, 30H			
5106	01	01480	ADD	A, C			
5107	CD0D50	01490	CALL	VIDEO			
510A	3E20	01500	LD	A, 20H			
510C	CD0D50	01510	CALL	VIDEO			
510F	C9	01520	RET				
		01530	M1	DEFM	'LOAD LEVEL-I TAPE IN CAS #2'		
		01540		DEFB	0DH		
		01550		DEFM	'LOAD LEVEL-II TAPE IN CAS #1'		
		01560		DEFB	0DH		
		01570		DEFM	'SET CAS #1 FOR RECORD & CAS #2 FOR PLAY'		
		01580		DEFB	0DH		
		01590		DEFM	'PRESS ENTER TO START COPYING OR'		
		01600		DEFB	0DH		
		01610		DEFM	'B TO RETURN TO LEVEL-2 BASIC'		
		01620	M2	DEFM	' INPUT ERROR '		
J13	50F5	J14	5064	LOOP	5027	VIDEO	500D VD 0033
J11	50E9	J14	5094	KEY	5002	RETB	1A19 KB 0020
J10	50D7	J3	5050	J1	5013	SYNC	0287 STACK 5300
J9	50D0	CHR	50E7	PRINT	509F	WRT	0264
J0	50C9	CAS	50C7	M1	5110	OUT	021E
M2	51AD	J2	5033	CLEAR	50A7	DEFC	0212
J5	5076	PNUM	50EF	BEG	5008	STOP	01F8

 FREE! A SHORT BASIC PROGRAM. (LEVEL ONE ONLY) INCLUDE A
 SELF-ADDRESSED, STAMPED ENVELOPE. G. FRANK HUMISTON
 946 S. ANZA #15, EL CAJON, CA 92020

"BUZZER"

10 A=0
 20 P.# A;
 30 G.20

You can exit this infinite loop via the "break" key. Could be used in games to accompany "you're shot down!", "you're sunk!", etc. No real harm is done to the reed relay with this routine.

FAST T-BUG RELOADER

Everett E. Ogden
175 Adams Street
Delmar, NY 12054

Those of you with Radio Shack's T-BUG program have probably been aggravated more than once by finding yourself unceremoniously dropped back into BASIC, or by getting locked up in a program and having to reset the machine. You then have to reload T-BUG, which takes about half a minute. The program is still there (well, most of it) but you can't get back to it without a cassette load. This change to T-BUG will allow you to do that, and repair the damage done by the BASIC monitor, by reloading just two bytes.

First it is necessary to understand how the computer knows it should run T-BUG after loading it, instead of returning to the BASIC monitor. There is nothing in the CLOAD routine to do this. It calls the subroutine CLOAD0 (which does the actual loading), performs a few other manipulations, and returns to the BASIC monitor. I had to cogitate on this a while (I'm new at this game) but I finally got it. I'll admit the people who designed this are pretty clever. When a subroutine is called, the return address is pushed onto the stack. Level I initializes its stack at 4200, and when CLOAD0 is called the return address is stored at 41FE and 41FF. BASIC programs start at 4200, but T-BUG runs from 4091 to 43B6 and overwrites the stack. It changes the return address to 40B1, which is the start of T-BUG.

If you get into BASIC you could return to T-BUG with a CLOAD of address 40B1 into the proper location. However, this leaves two problems. CLOAD0 loads a checksum after the last data byte (at 4200 in this case). Also, the BASIC monitor loads zeros from 409D to 40A6. These changes are repaired by the program below. It occupies memory space not used by T-BUG. Level I uses the space to store A\$ and B\$.

Load T-BUG, then make the following changes:

4070	ED 4B CB 43		; Data (same as 409D
4074	C3 BC 40		; to 40A6)
4077	CD 51 42		
407A	xx xx xx		; x = Don't Care
407D	01 0A 00	LD BC, 000A	; Repair 409D-40A6
4080	21 70 40	LD HL, 4070	
4083	11 9D 40	LD DE, 409D	
4086	ED B0	LDIR	
4088	21 00 42	LD HL, 4200	; Repair 4200
408B	36 32	LD M, 32	
408D	C3 B1 40	JP 40B1	; Start of T-BUG
41FE	7D 40		; New start address

Now punch 41FE to 41FF, preferably at the head of a leaderless tape to make it easy to find. Finally, punch 4070 to 43B6, which becomes your new T-BUG. If you find yourself in BASIC you need only rewind the cassette to the head and a CLOAD will put you back into T-BUG in seconds.

CPA wishes to purchase programs suitable for accountants.
ie- Form 1040 federal income tax and related schedules.
NYState & NYCity tax programs. General Ledger etc.
M.Mucciolo 635 Jericho Tpke. New Hyde Park, NY 11040

Tapetalk®

REGULAR FEATURES INCLUDE:

- * FORUM.
- * WHAT'S NEW
- * ME & MY 80
- * VOCAB.
- * SMALL BUSINESS ADVISOR
- * COMPTUTOR.
- * HOME IMPROVEMENT
- * TAPETEASERS™
- * GAME ROOM.

WIN	WIN	WIN	WIN	WIN	WIN	WIN	WIN	WIN
-----	-----	-----	-----	-----	-----	-----	-----	-----

CONTEST - GAMES - PRIZES

PRIZES - CONTEST - GAMES

WIN	WIN	WIN	WIN	WIN	WIN	WIN	WIN	WIN
-----	-----	-----	-----	-----	-----	-----	-----	-----

- * SCREEN PRINTER
- * LEVEL II (ROM)
 - * 16K (RAM)
- *** MORE

CASSETTE MAGAZINE
FOR THE

*
TRS-80

TAPE TALK IS A TRULY UNIQUE PUBLICATION FOR THE FIRST
GENERATION OF COMPUTER PIONEERS.

TAPE TALK IS A BI-MONTHLY 'MAGAZINE' EXCLUSIVELY FOR THE
TRS-80 SYSTEM.

TAPE TALK IS ALL ON CASSETTE--JUST LOAD IT IN.

A ONE YEAR SUBSCRIPTION IS ONLY \$30.00 AND IT INCLUDES
"AN INTRODUCTION TO TAPE TALK" CASSETTE.

THE "INTRODUCTION" OR SINGLE ISSUES MAY BE PURCHASED
INDIVIDUALLY FOR \$7.00 EACH.

MAIL COUPON OR CALL TOLL FREE -- 1-800/835-2246

PLEASE ENTER MY SUBSCRIPTION TO TapeTalk FOR:	
<input type="checkbox"/> 1 Year U.S. \$30.00	<input type="checkbox"/> 2 Years U.S. \$54.00
Make Check or Money Order payable to:	
TapeTalk, P. O. Box 54014, San Jose CA 95154	
Put on my <input type="checkbox"/> VISA/BankAmericard <input type="checkbox"/> Master Charge	
Card # _____	Expiration Date _____
Signature _____	
NAME (print) _____	
ADDRESS _____	
CITY _____	STATE _____ ZIP _____
NICKNAME: _____	
<input type="checkbox"/> 4 k	<input type="checkbox"/> 16 k <input type="checkbox"/> Level I <input type="checkbox"/> Level II

*A TRADEMARK OF RADIO SHACK

ANNOUNCEMENT!

THE TRS-80™ GETS ITS FIRST MAJOR INDEPENDENT PERIPHERAL

The things we all knew would happen have started. Basic Education Computers, Inc., a world leader in the field of audio-visual education, has interfaced its AVS-10 Early Learning System with the TRS-80 Home Computer.

THE AVS-10 EARLY LEARNING SYSTEM

As a stand-alone, the AVS-10 can teach a 3-year-old child to read *at the third grade level in just 15 weeks!* The system employs a unique phono-film cartridge which contains the audio and full-color visual training presentation on a single disk. Your child responds manually via a simple key-response system.

A MARRIAGE OF TECHNOLOGIES

Now, with the AVS-10 online to your TRS-80, the power of both machines is applied to advance your child's education. While the child operates and learns from the AVS-10, you monitor at the TRS-80 screen. A program running in the TRS-80 analyzes and reports on his progress, making suggestions on how to proceed. Later, the child transfers naturally to the TRS-80.

INEXPENSIVE RENTAL ARRANGEMENT

You can rent the AVS-10 complete with TRS-80 interface. For more information, call or write:

CONTRACT SERVICES ASSOCIATES

1846 W. Broadway

Anaheim, CA 92804

THOUGH OTHERS HAVE OFFERED THE SAME THING, I WISH TO OFFER THE FOLLOW-
SHIP A PROGRAM PRINTING SERVICE. DUE TO THE AVAILABILITY OF OTHER SUCH
SERVICES, I EXPECT TO BE ABLE TO OFFER ONE-DAY (ANY ANY) SERVICE. I CAN
PROCESS LEVEL I & II TAPES, AS WELL AS DISKS.

INSTRUCTIONS FOR THOSE INTERESTED:

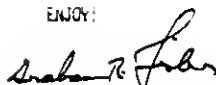
- 1) INCLUDE LEGIBLE INSTRUCTIONS AS TO
 - A) LEVEL OF BASIC (I OR II)
 - B) NUMBER OF PROGRAMS ON MEDIA (TAPE OR DISK)
 - C) WIDTH OF PRINT-OUT (5" TO 9" @ UP TO 132 COLUMNS)
 - D) IF SAMPLE RUNS ARE DESIRED (NOTE: I AM NOT OFFERING TO
DEBUG YOUR PROGRAMS:)
- 2) PACK ADEQUATELY!!!! YOUR TAPE OR DISK COULD BE RUINED SIMPLY
BY SPURIOUS MAGNETISM, SUCH AS HOBBY MAGNETS WHICH ARE REGULARLY
SENT THROUGH THE MAIL. TRY PACKING TAPES AND ESPECIALLY DISKS
IN A COOKIE TIN OR MINT TIN.
- 3) CHARGES ARE AS FOLLOWS
 - A) ONE CENT PER LINE OF PRINT-OUT
 - B) \$2.50 MINIMUM
 - C) \$1.25 FOR UNITED PARCEL SERVICE RETURN OF YOUR MATERIAL
 - D) NO CHARGE FOR FIRST CLASS RETURN, BUT ~~SEND STAMPER~~ SELF
ADDRESSED ENVELOPE (LARGE, MANILLA) FOR RETURN

A NOTE OF INTEREST TO LEVEL II DISK BASIC USERS... YOU CAN ~~QUICKLY~~ (BUT CERTAINLY)
DOUBLE THE USABLE STORAGE OF YOUR DISKS! MERELY CUT A NOTCH ~~OPPOSITE~~ CORRESPONDING
TO THE EXISTING NOTCH IN THE EDGE OF YOUR DISKS, EXACTLY OPPOSITE THE EXISTING
NOTCH, AND A PAIR OF HOLES EXACTLY OPPOSITE THE EXISTING HOLES, JUST OUTSIDE
THE CENTRAL DRIVE GRIPPER HOLE. IN OTHER WORDS, MAKE THE NOTCH ~~THE~~ A
MIRROR IMAGE OF THE TOP, AND THEN BACKUP THE FRONT OF THE DISK TO THE BACK.
VOILA! DOUBLE CAPACITY!

I ENCLOSE MY LISTING FOR A WORD PROCESSING SYSTEM WHICH IS QUITE WORKABLE,
ALLOWING MOST OF THE FEATURES OF THE IBM SYSTEM/32 WORD PROCESSING SYSTEM
TO BE EXECUTED! THE VARIABLES SEGMENTS ARE NOT INCLUDED DUE TO THE COMPLEXITY
OF THE SEGMENT AND A LITTLE GREED. THE PROGRAM IS ALREADY QUITE CAPABLE, AND
THE VARIABLES SEGMENT ALLOWS FOR TOTAL(!) AUTOMATION OF LETTER WRITING.
DIRECTIONS: FOLLOW THE PROMPTING OF THE PROGRAM, WHICH IS SELF-EXPLANATORY,
EXCEPT AS LISTED HERE. FOR LETTERS YOU DESIRE TO FORMAT THE WRITINGS OF,
USING THE LINE PRINTER - YOU MUST SPACE ONCE ON A LINE, OR THE PRINTER WILL
NOT CARRIAGE RETURN TO PUT SPACES VERTICALLY BETWEEN LINES. TO INSERT A
PROGRAM PROMPT FOR INFORMATION DURING THE PRINT SEGMENT, TYPE A < ## >
(DOUBLE AND SIGNS) FOLLOWED DIRECTLY BY THE INFORMATION NEEDED
- WITH NO SPACES - (AS: YOUR BALANCE IS PAID DUE, SHOWING A BALANCE
OF \$ACURRENT-BALANCE UNPAID). THE PROGRAM WILL SENSE THE < ## >, PRINT
"PLEASE ENTER THE 'CURRENT-BALANCE'" TO THE MONITOR SCREEN, WAIT FOR THE
DATA ENTRY, AND THEN INSERT THE INFORMATION IN THE LINE, PROCEEDING TO THE
END OF THE LETTER, OR THE NEXT < ## > CODE.

IT TAKES A LITTLE PLAYING WITH THE PROGRAM TO REALLY APPRECIATE IT. THE
SYSTEM WORKS WITH TAPE OR DISK (THE VARIABLES SEGMENT IS DISK-INTERACTIVE),
BUT DOES REQUIRE A LEVEL II MPU. THE SYSTEM PRINTS ROUGH OR FINISHED
DRAFTS, OR ALLOWS YOU TO SEE SEGMENTS OF THE LETTER ON THE MONITOR SCREEN.
IT ALLOWS EASY ENTRY OF NEW COPY, LESS EASY EDITING (I'M WORKING ON THIS),
• TRUE EASE OF STORAGE AND RETRIEVAL OF DATA, AND NEARLY AUTOMATIC OPERATION.
IT IS AVAILABLE ON DISK OR TAPE IN A MORE ADVANCED VERSION.

ENJOY!



GRAHAM R. FISHER
BOX 1227
CONCORD, NC 28025

P.S. THIS LETTER WAS PRODUCED BY MICROPRO.

P.P.S. I CANNOT BE RESPONSIBLE FOR PROGRAM MATERIAL PRINT-OUTS UNLESS GIVEN
TO UPS OF THE POST OFFICE AS I WILL NOT BE KEEPING ANY BRUNK COPIES OF
MATERIALS SENT TO ME FOR LISTING ON MY LINE PRINTER.

P.P.P.S. I'M AFRAID I FAILED TO MENTION THAT MICROPRO WORKS EXACTLY AS
WELL PREPARING CONTRACTS, LEGAL DRAFTS, PRE-PRINTED FORMS, OR JUST
ABOUT ANYTHING DESIRABLE. THANK YOU FOR YOUR INTEREST.

```

1 CLS PRINTAB(20). "*** WORDPROC ***" PRINT PRINT" A WORD-PROCESSING SYSTEM".
2 PRINT" THE TFS-80 MICROCOMPUTER SYSTEM" PRINT PRINTAB(45). "GRAHAM F FISHER"
3 PRINTAB(45). "BOX 1127" PRINTAB(45). "CONCORD, NC 28025" FOR I=1 TO 2500 NEXT I
4 CLEAR 12500
5 DIM LT$(120)
6 CLS
10 CLS PRINT"ENTER THE PROGRAM FUNCTION YOU WISH TO PERFORM"
12 PRINT" 1) INPUT A NEW LETTER"
14 PRINT" 2) SAVE THE LETTER CURRENTLY IN MEMORY TO DISK"
16 PRINT" 3) PRINT THE LETTER CURRENTLY IN MEMORY"
18 PRINT" 4) EDIT THE LETTER CURRENTLY IN MEMORY"
20 PRINT" 5) LOAD A LETTER FROM DISK"
26 PRINT" 6) PRINT ROUGH DRAFT"
33 PRINT" 7) ENTER VARIABLE LIST (TO MERGE W/LETTER)"
35 PRINT" 8) SAVE LETTER CURRENTLY IN MEMORY TO TAPE DRIVE #1"
37 PRINT" 9) LOAD A LETTER FROM CASSETTE DRIVE #1 "
55 PRINT" /) END THE PROGRAM"
60 F$=INKEY$ IF F$="" THEN 60 ELSE IF F$="/" THEN 30000 ELSE F=VAL(F$)
70 ON F GOSUB 120,500,5000,6000,7000,2000,1000,700,7200,30000
80 GOTO 10
120 CLS LNE=0.PRINT"PLEASE ENTER THE ROUGH COPY OF YOUR LETTER"
130 PRINT"ENTER $$$ WHEN YOUR LETTER IS COMPLETE"
150 LINE INPUT LT$(LNE)
160 IF MID$(LT$(LNE),1,3) = "$$$" THEN RETURN
170 LNE=LNE + 1
180 GOTO 150
500 CLOSE 1
505 INPUT"PLEASE ENTER YOUR LETTER NAME";NM$
510 OPEN "0",1,NM$
520 FOR I = 0 TO LNE
530 PRINT#1,LT$(I)
540 NEXT I
550 CLOSE 1
560 RETURN
700 REM THIS SUB SAVES LETTERS TO TAPE
710 CLS:PRINT:PRINT
720 PRINT"ENTER THE NAME OF THE LETTER YOU WISH TO SAVE TO TAPE,"
730 PRINT"FOLLOWING THESE RULES: 1) ONLY THE FIRST LETTER OF THE"
740 PRINT"LETTER NAME IS USED BY THE COMPUTER 2) NO LETTER ON ONE"
750 PRINT"TAPE MAY REPEAT THE FIRST LETTER OF THE NAME. END THE NAME"
760 PRINT"(NOT TO EXCEED 6 LETTERS) WITH A PERIOD. NOW ENTER THE"
770 PRINT"LETTER NAME:";
780 FOR I=1 TO 6
790 W$(I)=INKEY$ IF W$(I)=" " THEN 820
800 IF W$(I)="" THEN 790
810 NEXT I
820 CMD="T"
830 FOR I=0 TO LNE
840 PRINT#-1,LT$(I);", ";
850 NEXT I
860 CMD="R"
890 RETURN
1000 CLS PRINT"PLEASE ENTER THE VARIABLE FUNCTION YOU WISH TO PERFORM"
1005 PRINT" 1) ENTER THE VARIABLES ONE LETTER AT A TIME"
1010 PRINT" 2) ENTER ALL VARIABLES AT ONCE"
1015 PRINT" 3) ENTER VARIABLES FROM DISK FILE EN MASSE"
1020 PRINT" 4) CALL THE VARIABLES FROM DISK AS NEEDED"
1045 PRINT" 8) PRINT LETTER IN MEMORY & MERGE W/VARIABLES"
1050 PRINT" 9) ESCAPE VARIABLES SEGMENT"
1055 F$=INKEY$ IF F$="" THEN 1055 ELSE F=VAL(F$) IF F=9 THEN RETURN
1060 ON F GOSUB 1100,1200,1300,1400,1500,1000,1000,1000
1065 GOTO1000
1100 REM ENTER VAR 1 LETTER AT A TIME
1200 REM ENTER ALL VAR AT ONCE
1300 REM ENTER VARIABLES FROM DISK EN MASSE

```

```

1400 REM CALL VARIABLES FROM DISK AS NEEDED
1800 REM PRINT THE LETTER IN MEMORY, MERGE VARIABLES
1803 CLS FOR I=0 TO LNE-1
1807 FOR J=1 TO LEN(LT$(I))-1
1811 IF MID$(LT$(I),J,2)="&&" THEN 1903
1815 IF MID$(LT$(I),J,2)="**" THEN
1903 FOR K=J+2 TO LEN(LT$(I))
1907 IF MID$(LT$(I),K,1)=" " THEN 2000
1911 NEXT K
2000 LPRINT TAB(100);TIME$
2010 LPRINT " "
2020 FOR I=0 TO LNE-1
2030 LPRINT I;TAB(5);LT$(I)
2040 NEXT I
2085 LPRINT " "
2088 LPRINT TAB(100);TIME$
2090 RETURN
5000 CLS:FOR I = 0 TO LNE-1
5010 FOR J = 1 TO LEN(LT$(I))-1
5015 IF MID$(LT$(I),J,2) = "&&" THEN 5100
5020 NEXT J
5030 LPRINT LT$(I)
5040 GOTO 5800
5100 FOR K = J+2 TO LEN(LT$(I))
5110 IF MID$(LT$(I),K,1) = " " THEN 5200
5120 NEXT K
5200 PRINT"PLEASE ENTER THE ";MID$(LT$(I),J+2,K-(J+2))
5205 LINE INPUT IN$
5208 SL = LEN(LT$(I))
5210 OT$=( LEFT$(LT$(I),J-1) + IN$ + RIGHT$(LT$(I),SL-(K-1)))
5300 LPRINT OT$
5800 NEXT I
5820 PRINT"PRESS ANY KEY TO CONTINUE"
5840 X$=INKEY$
5842 IF X$="" THEN 5840
5846 RETURN
5850 INPUT"HIT 'ENTER' TO CONTINUE";R
5990 RETURN
6000 CLS
6010 PRINT"PLEASE ENTER THE EDIT FUNCTION "
6020 PRINT" 1) LIST A SECTION OF THE LINES"
6030 PRINT" 2) REPLACE A LINE"
6040 PRINT" 3) DELETE A LINE"
6050 PRINT" 4) INSERT A LINE"
6090 PRINT" 9) END EDITING
6100 F$=INKEY$:IF F$="" THEN 6100 ELSE F=VAL(F$)
6105 IF F=9 THEN RETURN
6110 ON F GOTO 6200,6300,6400,6500,6000,6000,6000,6000
6115 GOTO 6000
6200 PRINT" ENTER THE BEGINNING AND ENDING LINES"
6205 PRINT" THE NUMBER OF LINES IN THE LETTER IS ":LNE-1
6210 INPUT B,E
6215 IF (B<0) + (E<0) THEN PRINT"NO NEGATIVE NUMBERS PLEASE" GOTO 6200
6220 IF (B>LNE) + (E>LNE) THEN PRINT"LINE NUMBERS OUT OF RANGE" GOTO 6200
6230 FOR I = B TO E
6235 PRINT I,LT$(I)
6240 NEXT I
6242 FOR I = 1 TO 1000:NEXT I
6250 GOTO 6000
6300 PRINT"PLEASE ENTER THE LINE# TO REPLACE"
6310 INPUT N
6315 IF (N<0) + (N>LNE-1) PRINT"INVAL.ID NUMBER" GOTO 6300
6320 CLS:PRINT N,LT$(N)
6325 LINE INPUT LT$(N)
6330 GOTO 6000

```

```

6400 PRINT"PLEASE ENTER THE LINE YOU WISH TO DELETE"
6410 INPUT L
6420 IF (L<0) + (L>LNE+1) THEN PRINT"INVALID LINE#":GOTO 6400
6421 PRINT L:LT$(L)
6422 INPUT "ENTER YES TO VERIFY THAT THIS IS CORRECT",A$
6423 IF A$ <> "YES" THEN 6000
6425 FOR I = L+1 TO LNE
6430 LT$(I-1)= LT$(I)
6435 NEXT I
6440 LNE = LNE -1
6450 GOTO 6000
6500 PRINT"WHICH LINE WOULD YOU LIKE TO INSERT BEFORE"
6505 INPUT I
6510 IF (I<0) + (I>LNE-1) PRINT"INVALID LINE#":GOTO 6500
6520 FOR N = I TO LNE
6530 LT$(N+1) = LT$(N)
6540 PRINT"ENTER THE NEW LINE"
6545 LINE INPUT LT$(N)
6547 LNE=LNE+1
6550 GOTO 6000
7000 CLOSE1
7005 INPUT"PLEASE ENTER THE LETTER NAME",F$
7010 OPEN "I",1,F$
7020 LNE=0
7030 LINE INPUT#1,LT$(LNE)
7040 IF MID$(LT$(LNE),1,3) = "$$$" THEN RETURN
7050 LNE=LNE+1
7060 GOTO 7030
7200 REM THIS SUB LOADS LETTERS FROM TAPE INTO MEMORY ARRAY LT$
7210 PRINT"ENTER LETTER NAME, FOLLOWED BY A PERIOD THE LETTER NAME"
7220 PRINT"SHOULD NOT EXCEED SIX LETTERS"
7230 FOR I=1 TO 6
7240   W$(I)=INKEY$
7250   IF W$(I)=" " THEN 7280
7260   IF W$(I)="." THEN 7240
7270 NEXT I
7280 LNE=0
7290 CMD="T"
7300 INPUT#-1,LT$(LNE)
7310 IF MID$(LT$(LNE),1,3)="$$$" THEN 7340
7320 LNE=LNE+1
7330 GOTO 7300
7370 CMD="R"
7390 RETURN
30000 CLS:PRINT"SEE YOU LATER!!!!!"

```

TRS-80 OWNERS
 +HORSE RACE HANDICAPPING PROGRAM+
 LET YOUR COMPUTER MAKE \$\$ FOR YOU
 USING THIS TAPE AND A DAILY RACING FORM
 FANTASTIC RESULTS!!
 (PICK THE EXACT ORDER OF FINISH OF ALL
 HORSES IN THE BELMONT STAKES RACE)
 WRITTEN FOR 4K-LEVEL 1
 SEND \$11 FOR CASSETTE AND COMPLETE INSTRUCTIONS
 TO: HUFFMAN
 P.O. BOX 1566
 ASHLAND, KY 41101

INTRODUCES — — — THOUGHTWARE!

Designed To Make The Operation Of Your Personal Computer Easier And More Fun

● THE TRS-80 POKE GRAPHICS WORKSHEET

Supports quick, easy implementation of graphics in machine/assembly language or in fullest utilization of Level II Basic. — 1 pad of 25 sheets \$3.95
— 4 pads of 25 sheets \$6.95

● THE TRS-80 GRAPHICS WORKSHEET

Those desiring to continue the implementation of the SET and RESET commands are urged to order. — 2 pads of 25 sheets \$3.95
— 4 pads of 25 sheets \$6.95

● THE LINE MINDER

Tired of the ruler slipping off the Newsletter as you type in programs?
Set of 1-1x9 & 1-1x6 for 98¢

● OUR VARIABLE MINDER (It's The Handiest Darn Thing!)

See at a glance the availability of all 962 variables in Level II ... and their type assignments. Allows full implementation of up to 3848 for long-penciled programmers.
1 pad of 25 sheets \$3.95
2 pads of 25 sheets \$6.95

● AND OUR SELECTION OF NOTE PADS

Keep them handy to jot down those spur of the moment thoughts. Package of 6 pads (300 sheets, all same title) \$3.95

YOUR CHOICE OF TITLES BELOW, DON'T WRITE IT OFF —

(1) Bubble Babble, (2) Bus Lines, (3) Dandy Notes, (4) TRS-80 Owners Don't Baudot To Anyone, (5) Floppy Copy, (6) Padded Bits, (7) Under Disk-ussion, (8) Tandy Say — "Don't Make ASCII Of Self."

Illinois Residents Add 5% Sales Tax

JOHN REVELLE'S QUESTION ABOUT THE LINE PRINTER COMMANDS FOR PAGE SIZE ARE EASILY ANSWERED. THE POKE16424,## MUST BE FOLLOWED AT THE END OF THE PROGRAM WITH THE CHR\$(12) INSTRUCTION WHICH TELLS THE PRINTER TO 'GO TO THE TOP OF THE PAGE' THIS WILL CAUSE THE PRINTER TO FEED UP THE TOTAL NUMBER OF LINES FROM THE TOP OF THE PAGE TO THE BOTTOM AS IT IS INSTRUCTED TO DO IN THE 'POKE' COMMAND

MY PROBLEM WITH THE PRINTER IS THIS. TYPING ON THE SCREEN 'LPRINT' WILL NOT MAKE THE PRINTER PRINT IN ORDER FOR ME TO GET IT TO PRINT I MUST ENTER 'LPRINT' FOR EACH LINE AND THEN GIVE A 'RUN' COMMAND. IT WILL NOW PRINT BEAUTIFULLY. BUT THIS MEANS THAT WITHOUT THE PRINTER CONNECTED THE PROGRAM WON'T 'RUN'. IS THERE ANY ANSWER TO THIS OTHER THAN MAKING 2 TAPES? (ONE WITH AND ONE WITHOUT 'LPRINT')

YOU PEOPLE AT FAYETTEVILLE ARE DOING A GREAT JOB — ONE THAT RADIO SHACK OUGHT TO BE DOING AT LEAST SOME OF — TELLING US ABOUT THE MISTAKES IN THEIR MANUALS — KEEP UP THE GOOD WORK AND KEEP THE PROGRAMS, IDEAS, EXPERIENCES AND FRUSTRATIONS COMING.

REGARDS
PHIL REIDY
C. MINORNAI FUND
WORCESTER, MA 01609

DEAR TRS-80 FOLKS,

IN RESPONSE TO MR. REVELL'S LETTER IN THE AUGUST ISSUE, IT IS TRUE THAT 'LPRINT' BY ITSELF DOES NOTHING. HOWEVER UPON CALLING TANDY CUSTOMER SERVICE, I FOUND THAT THE CORRECT STATEMENT FOR A LINE FEED IS 'LPRINTCHR\$(138)'. AND SUGGEST THAT THOSE WITH LINE PRINTERS USE THIS INSTEAD OF THE 'LPRINT' '', AS IT DOES NOT CAUSE MOVEMENT OF THE PRINTING HEAD WHICH DOES CAUSE UNNECESSARY WEAR ON SAME.

MR. REVELL, IF YOU POKE16424,X WHERE 'X' IS ANY VALUE IT CHANGES THE NUMBER OF LINES PER PAGE AS IDENTIFIED BY THE LINE COUNTER AT ADDRESS 16425. THE IDEA IS THAT FOR INSTANCE, YOU WISH TO PRINT INFORMATION ON A LEGAL SIZED FORM (14'') AND THE PROPER STATEMENT WOULD BE 'POKE16424,84'. THEN AT THE END OF YOUR PROGRAM YOU WOULD INSERT THE COMMAND 'LPRINTCHR\$(12)' WHICH WOULD CAUSE A LINE FEED UNTIL THE COUNTER AT 16425 WAS EQUAL TO 84. PLEASE NOTE THAT IF INFORMATION WAS LPRINTED PAST THE 84TH LINE, THE LINE COUNTER WOULD AUTOMATICALLY RESET TO 0 AND WOULD LINE FEED UNTIL IT AGAIN REACHED 84.

AT FIRST, I FOUND IT DIFFICULT TO SET THE LINE LENGTH ON THE BACK OF THE LINE PRINTER WITH THE SMALL CONTROL. SO I SPENT ABOUT A DAY AND CALIBRATED IT, MAKING A TEMPLATE, AND INSTALLED A POINTER ON THE CONTROL. IT IS MUCH EASIER MERELY TO DIAL THE CHARACTERS PER LINE NEEDED INSTEAD OF GUESSING. IF ANYONE WITH A LINE PRINTER WOULD LIKE A PHOTOCOPY OF MY TEMPLATE, JUST WRITE AND ENCLOSE \$1.00 AND I WILL SEND SAME BACK TO YOU. IT MERELY FITS OVER THE CONTROL KNOB AND YOU CAN BUY A POINTER AT ANY RADIO SHACK OR JUST FROM YOUR SALVAGE BOX. SET THE POINTER ON THE CORRECT GAUGE ONCE AND NEVER HAVE TO GUESS AGAIN.

ALSO NOTE THAT THE LINE PRINTER ACTS JUST LIKE THE VIDEO SCREEN WHEN IT COMES TO THE 'TAB' STATEMENT. IN ORDER TO TAB PAST 63 I HAVE FOUND THE FOLLOWING USEFUL.

```
2 L=L-60 : LPRINTTAB(60);STRING$(L," "); : RETURN
```

```
99 L=120 : GOSUB2 : LPRINT X;
```

WHAT THE LINE PRINTER WILL NOW DO IS TO TAB OVER 60 SPACES, THEN PRINT A STRING OF BLANKS, THEN PRINT THE VALUE OF X AT THE 'L TAB' WHICH YOU SPECIFIED. BUT NOTE THAT EACH TIME THIS IS EXECUTED IN THE PROGRAM IT WILL ADD TO THE LINE LENGTH OF THE LINE PRINTER, I.E. IF LINE 99 IS RUN AGAIN BEFORE A END OF LINE IS EXECUTED THE LINE PRINTER HOLDS TAB(60), 60 SPACES, THE VALUE OF X, THEN UPON THE NEXT EXECUTION IT WILL ADD 60 MORE SPACES AND THE NEXT VALUE OF 'X' WHICH WILL EXCEED THE MAXIMUM LINE LENGTH OF 132 CHARACTERS. THE LINE PRINTER WILL TAB 60 SPACES, PRINT 60 BLANKS, PRINT THE VALUE OF X (SAY 10), AND IGNORE THE NEXT TAB(60), USE THE FIRST 9 BLANKS OF THE 'GOSUB CARRIAGE RETURN, PRINT 51 BLANKS, AND THEN THE SECOND VALUE OF X.

C. D. ROBERTSON, 2808 FORT WORTH NATIONAL BANK BUILDING
FORT WORTH, TEXAS 76102

* 16K PAM * 16K RAM * 16K RAM * 16K RAM * 16K RAM *

UP-GRADE YOUR TRS-80 TO 16K MEMORY OR ADD RAM TO YOUR EXPANSION INTERFACE. 16K RAM FOR ONLY \$139.95 OR 32K FOR ONLY \$255.00 + \$2.00 SHIPPING/HANDLING. SEND TO: TRS-80 USERS GROUP, 7554 SOUTHGATE RD. FAYETTEVILLE, NC 28304 (919) 867-5822
MAKE CHECK PAYABLE TO THE: TRS-80 USERS GROUP

This is a nonparametric test of independence or correlation coefficient which is very useful. One problem with the explosion in computing power is that an awful lot of tests are being used on non-normal data as if it were normal. In most cases "tau" is more reliable, robust and useful. This program calculates an approximation of a "z" score so that p can be read from a normal curve table (ie, if $z = 1.96$ the two-tailed p is .05).

I've used the Shell-Metzger sort to find the x column (its easier to write the program with this sort separately, and faster running time is enjoyed); but since I haven't had the time or wiadom to figure out how tau can be calculated I use the slightly slower merge sort described by Knight for sorting the y scores in each iteration. I didn't see the "butterfly" merge in the Jan-Feb CREATIVE COMPUTING in time to work it in; however, I also don't know how to calculate the statistic from the sort and its advantage over the Shell sort is not that significant for this program unless your N is enormous, in which case you probably don't want to use the TRS-80.

The program accomodates, automatically, for variations in N from column to column. As written it is set up to write the outcome into diskette, however a cassette routine is simple, and most applications will probably want to read off the screen. The first sort of each x/y takes about 2 minutes for N=100 and 1 minute for each subsequent y column.

Since I get a lot of spikes from the power supply, and have yet to get more than a 4 hour run on any matrix I've set it up so that you can specify a run starting with any column after the first; however, enter k-1 (same thing for the matrix cell, MC-1). You can terminate the run early by changing line 110 to IF KK< k (k being the last column you want to compare). DATA/READ STATEMENTS are somewhat cumbersome, but do save a lot of input time and storage space. I should also say that often I cannot get through a 42 column run for one row, and the feature enabling a start in the middle of the matrix is indispensable. I find lots of parentheses thrown in instead of commas, number changes and other glorious garbage which, fortunately, gives a SYNTAX ERROR message instead of running on to give you lousy data.

George von Hilsheimer, PhD
124 N. Clara Ave
DeLand, FL 32720
904-734-7664

```

5 REM * COPYRIGHT FRED BLECHMAN 1978 *
6 REM * 23958 ARCHWOOD ST. CANOGA PARK, CA 91307 *
7 REM * P M =3135 *
8 REM * SPIRANGLE PROGRAM *
10 CLS
20 X=62:Y=23
25 A=6:B=2
30 FOR Z=1 TO A
40 SET(X,Y)
45 X=X+1
46 IF X=123 GOTO 170
50 NEXT Z
60 FOR Z=1 TO B
70 SET(X,Y)
75 Y=Y+1
80 NEXT Z
85 C=A+6
90 For Z=1 to C
100 SET(X,Y)
105 X=X-1
110 NEXT Z
115 D=B+2
120 FOR Z= 1 TO D
130 SET(X,Y)
140 Y=Y-1
150 NEXT Z
155 H=D+6 R=D+2
160 GOTO 10
170 GOTO 170
180 END

```

```

300 PRINT"COMMAND"
310 X=10
320 GOSUB2900
330 T=2
340 GOSUB2300
350 RESTORE
360 READ Z$,T
370 IF Z$="###"THEN300
380 IF LEFT$(Z$,2)<LEFT$(T$,2)THEN360
390 ONT GOSUB500,700,800,850,900,1000,4600,5100,5000,5200,6000,4400,8000,5300
395 GOTO410
410 GOTO300

```

```

8000 REM ---LABELS---
8010 PRINT" LINES PER LABEL ";
8020 X=5
8030 GOSUB1900
8040 T=T-1
8045 I1=T
8050 IFT<0THEN8350
8060 FORI=0TOT
8070 PRINTX9$;" LINE ";I+1
8080 GOSUB2420
8090 V(I)=R
8100 NEXTI
8110 PRINT"POSITION FORMS. "
8120 PRINT"ENTER 1 FOR ALIGNMENT, 2 TO BEGIN"
8130 X=2
8140 GOSUB1900
8150 IFT=2THEN8220
8160 FORI=0TO5
8170 LPRINT"<----- ALIGNMENT ----->"
8180 NEXTI
8210 GOTO8130
8220 FORI=0TON
8230 GOSUB4100
8240 IFR<>0THEN8340
8250 FORI2=0TOI1
8260 IFV(I2)>M1THEN8290
8270 LPRINTMID$(R$(V(I2),I),1,30)
8280 GOTO8300
8290 LPRINTA(V(I2)-M1-1,I)
8300 NEXTI2
8305 IFI1>4THEN8340
8310 FORI2=0TO4-I1
8320 LPRINT CHR$(138)
8330 NEXTI2
8340 NEXTI
8350 PRINT"* * END OF LISTING * *"
8360 RETURN

```

HARD COPY NOW!!!

TELETYPE

THE USER'S GROUP IS EVALUATING TTEL-101 TO DRIVE A TELETYPE DIRECTLY FROM YOUR TRS-80 KEYBOARD. NO INTERFACE REQUIRED. SIMPLE 3 WIRE HOOK-UP AND YOU GET HARD COPY. ADAPTABLE TO MOST TELETYPES.

I N T E R E S T E D ?

DROP A LINE TO 808 LLOYD, AT THE GROUP ADDRESS.

DATA BASE * DATA BASE * DATA BASE * DATA BASE * DATA BASE

I HAVE A PROGRAM THAT I AM SURE YOU WILL LIKE. IT IS CALLED
DATABASE. I WILL SEND YOU THE PROGRAM ON TAPE OR DISK + A 120
PAGE BOOK FULL OF INFO ABOUT THE DATA BASE SYSTEM

THE COST FOR THE BOOK + TAPE OR DISK IS ONLY \$19.95
THE DATA BASE COMMANDS ARE LISTED BELOW:

ADD	COVARIANCE	NSORT
AVERAGE	DELETE	RESTORE
CLIST	LABELS	SUM
COMPRESS	LIST	VARIANCE

YOU DO NOT NEED TO KNOW PROGRAMMING. JUST LOAD THE PROGRAM
IN AND 'RUN'.

TABLE OF CONTENTS

SERIES INTRODUCTION - - - - -	-VI
ABOUT THIS VOLUME - - - - -	VII
PERMANENT STORAGE OF DATA BASE INFORMATION - -	VIII
1. THE DATA BASE CONCEPT - - - - -	1
2. THE BASIC STATEMENTS USED AND THE SYNTAX OF THEIR USE - - - - -	7
3. OBJECTIVES OF THE PEOPLE'S DATA BASE - - - - -	13
4. DATA BASE NUCLEUS - THE HEART OF PDB - - - - -	15
5. INTERMEDIATE COMMANDS - - - - -	57
6. HOW TO USE PDB - - - - -	95
7. APPLICATIONS - - - - -	95
A) BALANCING A CHECK BOOK	
B) ACCOUNTS SUMMARY	
C) INVENTORY	
D) GRADES	
E) PATIENT SCHEDULES	
F) JOB SCHEDULES	
G) PRIVATE LIBRARY	

READERS INFORMATION CARD - - - - - 123

DATA BASE * DATA BASE * DATA BASE * DATA BASE * DATA BASE

TRS-80 USERS GROUP

MY CHECK OR MONEY ORDER IS INCLUDED. PLEASE SEND ME A COPY OF POB
() ONE COPY + TAPE OR DISK - \$19.95 () TAPE () DISK
() SHIPPING/HANDLING - \$2.00

NAME ----- COMPANY -----

ADDRESS ----- TITLE -----

CITY ----- STATE ----- ZIP -----

EQUIPMENT: LEVEL I: () 4K () 16K LEVEL II: () 4K () 16K

() MINI DISK - NUMBER OF DRIVERS ----- () PRINTER -----

TRS-80 USERS GROUP
7554 SOUTHGATE RD.
FAYETTEVILLE, NC 283B4

TRS-80 USERS GROUP
7554 SOUTHGATE RD.
FRYETTEVILLE, NC 28304
(919) 867-5822

BULK RATE
U.S. POSTAGE
PAID
Permit No. 241
Fayetteville
N.C. 28303